CLAIMS

1. (currently amended) A method for executing a target application on a host processor comprising:

translating into host instructions each of a sequence of target instructions;

storing the translated host instructions, executing the stored host instructions; and

responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known; and

interpreting each target instruction in order from the point in execution at which correct state of a target processor is known.

2. (previously amended) The method of Claim 1 which further comprises:

collecting statistics regarding the execution of sequences of instructions which are interpreted.

3. (currently amended) A method for executing a target application on a host processor comprising the steps of:

executing host instructions representing each target instruction of the target application;

responding to an exception during execution of host instructions representing a target instruction by returning to a previous point in execution of the target application at which correct state of a target processor is known; and

thereafter executing host instructions by interpretation of the target instruction until the point of the exception.

4. (previously amended) A method as claimed in Claim 3 which further comprises collecting statistics regarding the execution of sequences of target instructions which are executed.

C1

5. (previously amended) A method as claimed in Claim 4 in which the statistics include the number of times the sequence of target instructions have executed.

6. (previously amended) A method as claimed in Claim 4 in which the statistics include address of an instruction to which a target instruction including a branch operation branches.

7. (previously amended) A method as claimed in Claim 4 in which the statistics include a likelihood of a branch being taken.

8. (currently amended) A system for executing a target application designed for execution on a target processor on a host processor having an instruction set different than that of the target processor comprising:

means for translating sequences of target instructions and storing each translated sequence of instructions,

means for selecting a stored translated sequence of instructions for execution,

means for responding to an exception during execution of a stored translated instruction by rolling back to a previous point in execution at which correct state of a target processor is known, and

means for interpreting each target instruction in order from a point in execution at which correct state of a target processor is known through the target instruction causing the exception.

9. (previously amended) A system as claimed in Claim 8 in which the means for interpreting is an interpreter software executing on the host processor, and

the means for translating is dynamic translation software executing on the host processor.

Claim 10. (previously withdrawn)

Claim 11. (previously withdrawn)

Claim 12. (previously withdrawn)

Claim 13. (previously withdrawn)

14. (previously added) The method of Claim 1, wherein the exception results from speculative execution of a branch instruction of the target application.

15. (previously added) The method of Claim 1,

16. (previously added) The method of Claim 1, further comprising:

collecting statistics regarding the execution of sequences of target instructions which are executed.

17. (previously added) The method of Claim 16, wherein the statistics include the number of times a branch of target instructions have executed.

18. (previously added) The method of Claim 17, further comprising:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.

19. (previously added) The system of Claim 8, further comprising:

collecting statistics including the number of times a branch of target instructions have executed.

20. (previously added) The system of Claim 19, further comprising:

speculatively translating target instructions into host instructions based on a likelihood of a branch being taken.